Structured Programming Language (CSE113)
Lecture # 01
Introduction to C Programming

C Programming Language

C is a powerful general-purpose programming language. It is fast, portable and available in all platforms. It is used for wide range of applications from Operating systems like **Windows** and **iOS** to **software** that is used for creating **3D movies**.

C programming is considered as the base for other programming languages, that is why it is known as mother language.

It can be defined by following ways:

- Mother language
- System programming language
- Procedure-oriented programming language
- Structured programming language
- Mid-level programming language

C is called a **structured programming language** because to solve a large problem, C programming language divides the problem into smaller modules called **functions** or **procedures** each of which handles a particular responsibility. The program which solves the entire problem is a collection of such **functions**.

History of C Language

- C programming language was developed in 1972 by **Dennis Ritchie** at bell laboratories of AT&T (American Telephone & Telegraph), located in U.S.A.
- C programming language features were derived from an earlier language called “B” (Basic Combined Programming Language – BCPL)
- C language was invented for implementing **UNIX** operating system

Uses of C programming language:

The C programming language is used for developing system applications that forms a major portion of operating systems such as Windows, UNIX and Linux. Below are some examples of C being used.

- Database systems
- Graphics packages
- Word processors
- Spreadsheets
- Operating system development
- Compilers and Assemblers
- Network drivers
- Interpreters

First C program: “Hello, World!” program:

“Hello, World!” is a simple program that displays “Hello, World!” on the screen. Since, it’s a very simple program, it is used to illustrate the basic syntax of any programming language.

```
#include <stdio.h>
int main()
{
   printf("Hello World!");
   return 0;
}
```

Let’s look into various parts of the above C program:

- **#include <stdio.h>** includes the **standard input output library functions**. The **printf()** function is defined in **stdio.h**.
- The **main()** function is the **entry point of every program** in C language. The code inside the curly braces { } is the body of main() function. The main() function is mandatory in every C program.
- The **printf()** is a **library function** is used for output. It prints the given statement to the console. In this program, it displays **Hello World!** on the screen.
- The return statement **return 0;** inside the main() function ends the program. This statement isn’t mandatory. However, it’s considered good programming practice to use it.

printf() & scanf() in C

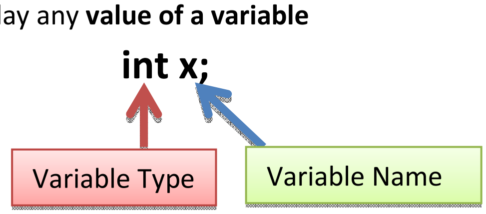
The **printf()** and **scanf()** functions are used for **output** and **input** in C language. Both functions are built-in library functions, defined in **stdio.h** (header file).

printf() function: The printf() function is used for output. It prints the given statement to the console.

scanf() function: The scanf() function is used for input. It reads the input data from the console.

Purposes of printf() function:

1. To display any **message**
   Example: printf(“Welcome 2017”);
2. To create **new line**, **tab space** etc.
   Example: printf(“\n”);
   printf(“Hello\n2017”);
   printf(“\t”);
3. To display any **value**
   Example: printf(“%d”,10+20+6);
   %d for **integer** **%f** for **float** **%lf** for **double** **%c** for **char** (character)
4. To display any **value of a variable**



Example:

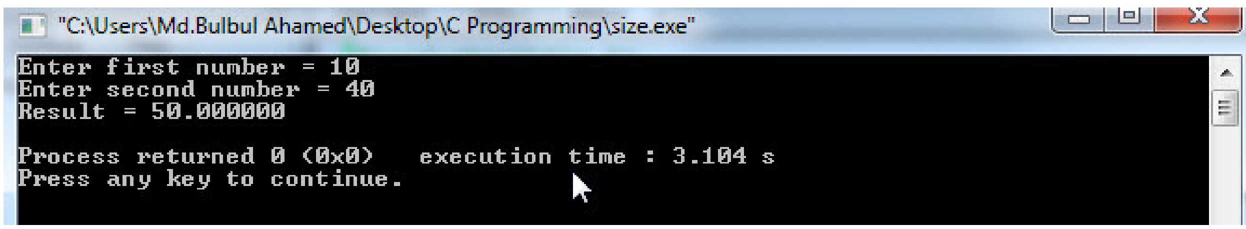
```
int x = 100;   // x is a variable name,   int is variable type
printf(“%d”, x);    // %d for integer
```

Program Example: Write a C program to **add two numbers**. Here, **variable type** should be **float** and **value of variables** must be taken **using keyboard**.

<u>Source Code:</u>

```
#include<stdio.h>
int main()
{
   float x, y; //Variable declaration
   printf("Enter first number = "); //message for user
   scanf("%f", &x); //input from user
   printf("Enter second number = "); //message for user
   scanf("%f", &y); //input from user
   printf("Result = %f\n", x + y); //output
   return 0;
}
```

<u>Output:</u>



Comments in C:

Comments in C language are used to provide information about lines of code. It is widely used for documenting code. Comments are explanatory notes for user. They are ignored by the compiler. There are two types of comments:

1. Single line comment
   Example: //Creator: Bulbul Ahamed
2. Multiple line comment
   Example:
   ```
   /* Creator: Bulbul Ahamed
      Date: February 12, 2017 */
   ```

Variables in C:

A **variable** is a name of memory location. It is used to store data. Its value can be changed and it can be reused many times.

The **syntax** to declare a variable:

type variable_list;

The example of declaring variable is given below:

- **int** a;
- **float** b;
- **char** c;

Here, **a, b, c** are **variables** and **int**, **float**, **char** are **data types**.

We can also provide values while declaring the variables as given below:

- **int** a = 10, b = 20; //declaring 2 variable of integer type
- **float** f = 20.8;
- **char** c = 'A';

Rules for defining variables

- A variable can have **alphabets**, **digits** and **underscore**.
- A **variable name can start** with **alphabet** and **underscore** only. It can't start with digit.
- **No white space** is allowed within variable name.
- A variable name must not be any **reserved word** or **keyword** e.g. int, float etc.

Valid variable names:

- ✓ int **a**;
- ✓ int **_ab**;
- ✓ int **a30**;

Invalid variable names:

- ✓ int **2**;
- ✓ int **a b**;
- ✓ int **long**;

Types of Variables in C

There are many types of variables in c:

1. local variable
2. global variable
3. static variable
4. automatic variable
5. external variable

Local Variable : A variable that is declared inside the function or block is called local variable.

```
void main() {
    int x = 10;   //local variable
}
```

You must have to initialize the local variable before it is used.

Global Variable: A variable that is declared outside the function or block is called global variable. Any function can change the value of the global variable. It is available to all the functions.

```
int value = 20; //global variable
void main () {
   int x = 10; //local variable
}
```

Data Types in C

A data type specifies the type of data that a variable can store such as integer, floating, character etc. There are 4 types of data types in C language.

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Void Data Type	void
Enumeration Data Type	enum

The following table provides the details of standard integer types with their storage sizes and value ranges –

Types	Storage size
char	1 byte
int	2 or 4 bytes
float	4 bytes
double	8 bytes

To get the exact size of a type or a variable on a particular platform, you can use the **sizeof** operator. The expressions **sizeof(type)** yields the storage size of the object or type in bytes.

Program Example: sizeof() function

```
#include<stdio.h>
int main() {
   printf("%d\n", sizeof(char));
   printf("%d\n", sizeof(int));
   printf("%d\n", sizeof(float));
   printf("%d\n", sizeof(double));
   return 0;
}
```

Keywords in C

A **keyword** is a reserved word. You cannot use it as a variable name, constant name etc. There are only 32 reserved words (keywords) in C language. A list of 32 keywords in c language is given below:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

Escape sequences in C

The backslash *(\)* is called an escape character. It indicates that ***printf()*** is supposed to do something out of the ordinary.

List of common Escape Sequences in C:

Escape Sequence	**Meaning**
\n	New Line
\t	Tab (Horizontal)
\'	Single Quote
\"	Double Quote
\?	Question Mark
\a	Alarm or Beep
\0	Null

Constants in C

A constant is a value or variable that can't be changed in the program, for example: 10, 'a', 3.4, "programming" etc. There are different types of constants in C programming.

Constant	Example
Decimal Constant	10, 20, 450 etc.
Real or Floating-point Constant	10.3, 20.2, 450.6 etc.
Character Constant	'a', 'b', 'x' etc.
String Constant	"c", "program", "Hello World" etc.

There are **two ways to define constant** in C programming.

1. **const** keyword
2. **#define** preprocessor

Example:

const float PI=3.1416;

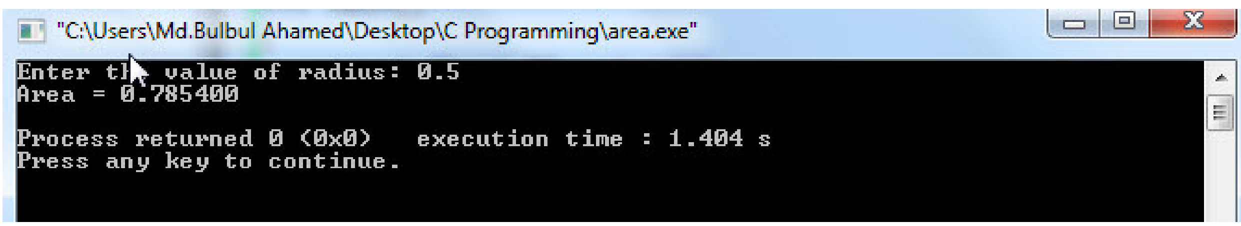
#define PI 3.1416

Program Example: A program to find the area of a circle. Here, the value of radius must be taken from user.

Source Code:
```
#include<stdio.h>
#define PI 3.1416
int main()
{
   float radius,area;
   printf("Enter the value of radius: ");
   scanf("%f",&radius);
   area = PI * radius * radius;
   printf("Area = %f\n",area);
}
```

Source Code:
```
#include<stdio.h>
int main()
{
   const float PI = 3.1416;
   float radius,area;
   printf("Enter the value of radius: ");
   scanf("%f",&radius);
   area = PI * radius * radius;
   printf("Area = %f\n",area);
}
```

Output:



....